

```

AAAAAAAAA  NNN      NNN      AAAAAAAAA  LLL      YYY      YYY  ZZZZZZZZZZZZZZZZ
AAAAAAAAA  NNN      NNN      AAAAAAAAA  LLL      YYY      YYY  ZZZZZZZZZZZZZZZZ
AAAAAAAAA  NNN      NNN      AAAAAAAAA  LLL      YYY      YYY  ZZZZZZZZZZZZZZZZ
AAA        AAA  NNN      NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNN      NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNN      NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNNNNN   NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNNNNN   NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNNNNN   NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNN      NNN  NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNN      NNN  NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNN      NNN  NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAAAA  LLL      YYY      YYY  ZZZ
AAAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAAAA  LLL      YYY      YYY  ZZZ
AAAAAAAAAAAAAAAA  NNN      NNNNNN  AAAAAAAAAAAAAAAAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNN      NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNN      NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNN      NNN      AAA        AAA  LLL      YYY      YYY  ZZZ
AAA        AAA  NNN      NNN      AAA        AAA  LLLLLLLLLLLLLLLLL  YYY      ZZZZZZZZZZZZZZZZ
AAA        AAA  NNN      NNN      AAA        AAA  LLLLLLLLLLLLLLLLL  YYY      ZZZZZZZZZZZZZZZZ
AAA        AAA  NNN      NNN      AAA        AAA  LLLLLLLLLLLLLLLLL  YYY      ZZZZZZZZZZZZZZZZ

```

```
000000 88888888 JJ MM MM 111111 SSSSSSSS CCCCCCCC
000000 88888888 JJ MM MM 111111 SSSSSSSS CCCCCCCC
00 00 88 88 JJ JJ MMMM MMMM II II SS SS CC CC
00 00 88 88 JJ JJ MMMM MMMM II II SS SS CC CC
00 00 88 88 JJ JJ MM MM II II SS SSSSSS CC CC
00 00 88888888 JJ JJ MM MM II II SS SSSSSS CC CC
00 00 88888888 JJ JJ MM MM II II SS SSSSSS CC CC
00 00 88 88 JJ JJ MM MM II II SS SS CC CC
00 00 88 88 JJ JJ MM MM II II SS SS CC CC
00 00 88 88 JJ JJ MM MM II II SS SS CC CC
00 00 88 88 JJ JJ MM MM II II SS SS CC CC
000000 88888888 JJJJJJ MM MM 111111 SSSSSSSS CCCCCCCC
000000 88888888 JJJJJJ MM MM 111111 SSSSSSSS CCCCCCCC

LL 111111 SSSSSSSS
LL 111111 SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL 111111 SSSSSSSS
LLLLLLLLLL 111111 SSSSSSSS
```

```

0001 0 title 'OBJMISC - Analyze Miscellaneous Object Records'
0002 0 module objmisc (
0003 1 ident='V04-000') = begin
0004 1
0005 1
0006 1 *****
0007 1 *
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 * ALL RIGHTS RESERVED.
0011 1 *
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 * TRANSFERRED.
0018 1 *
0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 * CORPORATION.
0022 1 *
0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *****
0027 1
0028 1
0029 1
0030 1 **
0031 1 Facility: VAX/VMS Analyze Facility, Analyze Miscellaneous Object Records
0032 1
0033 1 Abstract: This module is responsible for analyzing the following object
0034 1 record types:
0035 1 EOM End-of-Module Records
0036 1 HDR Header Records
0037 1 LNK Link Option Records
0038 1 and also reserved record types
0039 1
0040 1
0041 1 Environment:
0042 1
0043 1 Author: Paul C. Anagnostopoulos, Creation Date: 13 January 1981, my birthday!
0044 1
0045 1 Modified By:
0046 1
0047 1 V03-004 ROP0020 Robert Posniak 11-JUL-1984
0048 1 Ensure we don't point beyond header record after
0049 1 we print creation date/time.
0050 1
0051 1 V03-003 MCN0158 Maria del C. Nasr 22-Mar-1984
0052 1 Add size parameter to call to ANL$CHECK_SYMBOL, since now
0053 1 it can be up to 39 characters (maximum size of shareable image
0054 1 name).
0055 1
0056 1 V03-002 JWT0122 Jim Teague 26-May-1983
0057 1 Remove requirement for a patch date/time field. Such
  
```


OBJMISC
V04-000

OBJMISC - Analyze Miscellaneous Object Records

E 1
15-Sep-1984 23:42:42
14-Sep-1984 11:52:57

VAX-11 Bliss-32 V4.0-742
[ANALYZ.SRC]OBJMISC.B32;1

Page 2
(1)

: 58
: 59
: 60
: 61
: 62
: 63
: 64

0058 1 :
0059 1 :
0060 1 :
0061 1 :
0062 1 :
0063 1 :
0064 1 :--

a field is meaningless, and the Linker ignores it.

V03-001 PCA1011

Paul C. Anagnostopoulos 1-Apr-1983

Change the message prefix to ANLOBJ\$ to ensure that
message symbols are unique across all ANALYZEs. This
is necessitated by the new merged message files.

```

: 66      0065 1 %sbttl 'Module Declarations'
: 67      0066 1
: 68      0067 1  Libraries and Requires:
: 69      0068 1
: 70      0069 1
: 71      0070 1  library 'lib';
: 72      0071 1  require 'objexereq';
: 73      0507 1
: 74      0508 1
: 75      0509 1  Table of Contents:
: 76      0510 1
: 77      0511 1
: 78      0512 1  forward routine
: 79      0513 1      anl$object_eom: novalue,
: 80      0514 1      anl$object_hdr: novalue,
: 81      0515 1      anl$object_hdr_mhd: novalue,
: 82      0516 1      anl$object_record_size: novalue,
: 83      0517 1      anl$object_hdr_text: novalue,
: 84      0518 1      anl$object_hdr_mtc: novalue,
: 85      0519 1      anl$object_lnk: novalue;
: 86      0520 1
: 87      0521 1
: 88      0522 1  External References:
: 89      0523 1
: 90      0524 1
: 91      0525 1  external routine
: 92      0526 1      anl$check_flags,
: 93      0527 1      anl$check_symbol,
: 94      0528 1      anl$check_when,
: 95      0529 1      anl$format_error,
: 96      0530 1      anl$format_flags,
: 97      0531 1      anl$format_hex,
: 98      0532 1      anl$format_line,
: 99      0533 1      anl$object_env_check,
: 100     0534 1      anl$object_psect_check,
: 101     0535 1      anl$object_psect_ref,
: 102     0536 1      anl$object_record_line,
: 103     0537 1      anl$object_tir_clean,
: 104     0538 1      anl$report_line;
: 105     0539 1
: 106     0540 1
: 107     0541 1  Own Variables:
: 108     0542 1
: 109     0543 1  The following variable is used to remember the record size from
: 110     0544 1  the module header.
: 111     0545 1
: 112     0546 1  own
: 113     0547 1      mhd_record_size: long initial(obj$cm_maxrecsiz);
```

```
115 0548 1 %sbttl 'ANL$OBJECT_EOM - Analyze EOM and EOMW Records'
116 0549 1 ++
117 0550 1 Functional Description:
118 0551 1 This routine analyzes end of module records, of which there are
119 0552 1 two flavors.
120 0553 1
121 0554 1 Formal Parameters:
122 0555 1 record_number Number of this object record.
123 0556 1 the_record Address of descriptor of the record.
124 0557 1
125 0558 1 Implicit Inputs:
126 0559 1 global data
127 0560 1
128 0561 1 Implicit Outputs:
129 0562 1 global data
130 0563 1
131 0564 1 Returned Value:
132 0565 1 none
133 0566 1
134 0567 1 Side Effects:
135 0568 1
136 0569 1 --
137 0570 1
138 0571 1
139 0572 2 global routine anl$object_eom(record_number,the_record): novalue = begin
140 0573 2
141 0574 2 bind
142 0575 2 record_dsc = .the_record: descriptor;
143 0576 2
144 0577 2 own
145 0578 2 transfer_flags_def: vector[2,long] initial(
146 0579 2 0,
147 0580 2 uplit byte (%ascic 'EOM$V_WKTFR')
148 0581 2 );
149 0582 2
150 0583 2 local
151 0584 2 status: long,
152 0585 2 scanp: ref block[,byte],
153 0586 2 fit_ok: byte;
154 0587 2
155 0588 2 builtin
156 0589 2 nullparameter;
157 0590 2
158 0591 2
159 0592 2 ! If we are called with no arguments, it means that we reached the end of
160 0593 2 ! an object file and were missing an end-of-module record. In this case,
161 0594 2 ! we are to "force" and end-of-module. Skip all the record analysis stuff.
162 0595 2
163 0596 3 if not nullparameter(1) then (
164 0597 3
165 0598 3 ! First we print a major line for the record. We won't indent this code
166 0599 3 ! because it is so long.
167 0600 3
168 0601 3 scanp = .record_dsc[ptr];
169 0602 4 anl$object_record_line((if .scanp[obj$b_rectyp] eqlu obj$c_eom then anl$obj$_objeomrec
170 0603 3 else anl$obj$_objeomwrec),
171 0604 3 .record_number,record_dsc);
```



```
172 0605 3 anl$report_line(0);
173 0606 3
174 0607 3 ! Now we make sure the severity is present and print it.
175 0608 3
176 0609 3 fit_ok = true;
177 0610 3
178 0611 3 ensure_field_fit(eom$b_comcod,record_dsc);
179 0612 4 if .fit_ok then (
180 0613 5     anl$format_line(0,1, (selectoneu .scanp[eom$b_comcod] of set
181 0614 5         [eom$c_success]: anlobj$_objeomsevsuc;
182 0615 5         [eom$c_warning]: anlobj$_objeomsevwrn;
183 0616 5         [eom$c_error]: anlobj$_objeomseverr;
184 0617 5         [eom$c_abort]: anlobj$_objeomsevabt;
185 0618 5         [4 to 10]: anlobj$_objeomsevres;
186 0619 5         [otherwise]: anlobj$_objeomsevign;
187 0620 4         tes),
188 0621 4         .scanp[eom$b_comcod]);
189 0622 4     if .scanp[eom$b_comcod] gequ 4 and .scanp[eom$b_comcod] lequ 10 then
190 0623 4         anl$format_error(anlobj$_objeombadsev);
191 0624 3 );
192 0625 3
193 0626 3 ! Now we are done if that is the end of the record.
194 0627 3
195 0628 4 if .record_dsc[len] gtru 2 then (
196 0629 4
197 0630 4     ! I guess we have a transfer address. First there is a psect number,
198 0631 4     ! which is either a byte or word depending on the record type. Be sure
199 0632 4     ! to record the reference.
200 0633 4
201 0634 5     if .scanp[obj$b_rectyp] eqlu obj$c_eom then (
202 0635 5         ensure_field_fit(eom$b_psindx,record_dsc);
203 0636 6         if .fit_ok then (
204 0637 6             anl$format_line(0,1,anlobj$_objpsect,.scanp[eom$b_psindx]);
205 0638 6             anl$object_psect_ref(.scanp[eom$b_psindx]);
206 0639 6             scanp = scanp[eom$l_tfradr];
207 0640 5         );
208 0641 5
209 0642 5     ) else (
210 0643 5
211 0644 5         ensure_field_fit(eomw$b_psindx,record_dsc);
212 0645 6         if .fit_ok then (
213 0646 6             anl$format_line(0,1,anlobj$_objpsect,.scanp[eomw$b_psindx]);
214 0647 6             anl$object_psect_ref(.scanp[eomw$b_psindx]);
215 0648 6             scanp = scanp[eomw$l_tfradr];
216 0649 5         );
217 0650 4     );
218 0651 4
219 0652 4     ! Now we have the transfer offset itself. Print it.
220 0653 4
221 0654 4     ensure_field_fit(0,0,32,0,record_dsc);
222 0655 5     if .fit_ok then (
223 0656 5         anl$format_line(0,1,anlobj$_objvalue,.scanp[0,0,32,0]);
224 0657 5         if .scanp[0,0,32,0] gtru %x*3ffffff then
225 0658 5             anl$format_error(anlobj$_objp0space);
226 0659 5         scanp = .scanp + 4;
227 0660 4     );
228 0661 4
```

OBJMISC
V04-000

OBJMISC - Analyze Miscellaneous Object Records 15-Sep-1984 23:42:42 VAX-11 Bliss-32 V4.0-742
ANL\$OBJECT_EOM - Analyze EOM and EOMW Records 14-Sep-1984 11:52:57 [ANALYZ.SRC]OBJMISC.B32;1

Page 6
(3)

```
: 229      0662  4      ! Again, the record may end at this point.  If so, we are done.
: 230      0663  4
: 231      0664  5      if .record_dsc[ptr]+.record_dsc[len] gtru .scanp then (
: 232      0665  5
: 233      0666  5          ! OK, so there must be the transfer flags byte.
: 234      0667  5          ! Print it and check it.
: 235      0668  5
: 236      0669  5          anl$format_flags(1,anlobj$_objeomflags,.scanp[0,0,8,0],transfer_flags_def);
: 237      0670  5          anl$check_flags(.scanp[0,0,8,0],transfer_flags_def);
: 238      0671  5          increment(.scanp);
: 239      0672  5
: 240      0673  5          ! We must ensure that there are no spurious bytes at the end.
: 241      0674  5
: 242      0675  5          if .record_dsc[ptr]+.record_dsc[len] gtru .scanp then
: 243      0676  5              anl$format_error(anlobj$_extrabytes);
: 244      0677  4      );
: 245      0678  3  );
: 246      0679  2  );
```



```

. EXTRN  ANLOBJ$_OK, ANLOBJ$_ANYTHING
. EXTRN  ANLOBJ$_DATATYPE
. EXTRN  ANLOBJ$_ERRORCOUNT
. EXTRN  ANLOBJ$_ERRORNONE
. EXTRN  ANLOBJ$_ERRORS, ANLOBJ$_EXEFIXA
. EXTRN  ANLOBJ$_EXEFIXAIMAGE
. EXTRN  ANLOBJ$_EXEFIXALINE
. EXTRN  ANLOBJ$_EXEFIXCOUNT
. EXTRN  ANLOBJ$_EXEFIXEXTRA
. EXTRN  ANLOBJ$_EXEFIXFIXED
. EXTRN  ANLOBJ$_EXEFIXFLAGS
. EXTRN  ANLOBJ$_EXEFIXG
. EXTRN  ANLOBJ$_EXEFIXGIMAGE
. EXTRN  ANLOBJ$_EXEFIXGLINE
. EXTRN  ANLOBJ$_EXEFIXLIST
. EXTRN  ANLOBJ$_EXEFIXNAME
. EXTRN  ANLOBJ$_EXEFIXNAME0
. EXTRN  ANLOBJ$_EXEFIXP
. EXTRN  ANLOBJ$_EXEFIXPSECT

```

```
.EXTRN ANLOBS$-EXEFIXUP
.EXTRN ANLOBS$-EXEFIXUPNONE
.EXTRN ANLOBS$-EXEGST, ANLOBS$-EXEHDR
.EXTRN ANLOBS$-EXEHDRACTIVE
.EXTRN ANLOBS$-EXEHDRBLKCOUNT
.EXTRN ANLOBS$-EXEHDRCHANCOUNT
.EXTRN ANLOBS$-EXEHDRCHANDEF
.EXTRN ANLOBS$-EXEHDRDECECO
.EXTRN ANLOBS$-EXEHDRDMT
.EXTRN ANLOBS$-EXEHDRDST
.EXTRN ANLOBS$-EXEHDRFILEID
.EXTRN ANLOBS$-EXEHDRFIXED
.EXTRN ANLOBS$-EXEHDRFLAGS
.EXTRN ANLOBS$-EXEHDRGBLIDENT
.EXTRN ANLOBS$-EXEHDRGST
.EXTRN ANLOBS$-EXEHDRIDENT
.EXTRN ANLOBS$-EXEHDRIMAGEID
.EXTRN ANLOBS$-EXEHDRISD
.EXTRN ANLOBS$-EXEHDRISDBASE
.EXTRN ANLOBS$-EXEHDRISDCOUNT
.EXTRN ANLOBS$-EXEHDRISDFlags
.EXTRN ANLOBS$-EXEHDRISDGBLNAME
.EXTRN ANLOBS$-EXEHDRISDNUM
.EXTRN ANLOBS$-EXEHDRISDPFCDEF
.EXTRN ANLOBS$-EXEHDRISDPFCsiz
.EXTRN ANLOBS$-EXEHDRISDTYPE
.EXTRN ANLOBS$-EXEHDRISDVBN
.EXTRN ANLOBS$-EXEHDRLINKID
.EXTRN ANLOBS$-EXEHDRMATCH
.EXTRN ANLOBS$-EXEHDRNAME
.EXTRN ANLOBS$-EXEHDRNOPATCH
.EXTRN ANLOBS$-EXEHDRPAGECOUNT
.EXTRN ANLOBS$-EXEHDRPAGEDEF
.EXTRN ANLOBS$-EXEHDRPATCH
.EXTRN ANLOBS$-EXEHDRPATCHDATE
.EXTRN ANLOBS$-EXEHDRPRIV
.EXTRN ANLOBS$-EXEHDRROPATCH
.EXTRN ANLOBS$-EXEHDRRWPATCH
.EXTRN ANLOBS$-EXEHDRSYMDBG
.EXTRN ANLOBS$-EXEHDRSYSVER
.EXTRN ANLOBS$-EXEHDRTEXTVBN
.EXTRN ANLOBS$-EXEHDRTIME
.EXTRN ANLOBS$-EXEHDRTYPEEXE
.EXTRN ANLOBS$-EXEHDRTYPEELIM
.EXTRN ANLOBS$-EXEHDRUSERECO
.EXTRN ANLOBS$-EXEHDRXFER1
.EXTRN ANLOBS$-EXEHDRXFER2
.EXTRN ANLOBS$-EXEHDRXFER3
.EXTRN ANLOBS$-EXEHDRHEADING
.EXTRN ANLOBS$-EXEPATCH
.EXTRN ANLOBS$-FLAG, ANLOBS$-HEXDATA
.EXTRN ANLOBS$-HEXHEADING1
.EXTRN ANLOBS$-HEXHEADING2
.EXTRN ANLOBS$-INDMSGSEC
.EXTRN ANLOBS$-INTERACT
.EXTRN ANLOBS$-MASK, ANLOBS$-OBJCPRREC
.EXTRN ANLOBS$-OBJDBGREC
```

```
.EXTRN ANLOBS$OBJENV, ANLOBS$_OBJEOMFLAGS
.EXTRN ANLOBS$_OBJEOMREC
.EXTRN ANLOBS$_OBJEOMSEVABT
.EXTRN ANLOBS$_OBJEOMSEVERR
.EXTRN ANLOBS$_OBJEOMSEVIGN
.EXTRN ANLOBS$_OBJEOMSEVRES
.EXTRN ANLOBS$_OBJEOMSEVSUC
.EXTRN ANLOBS$_OBJEOMSEVWRN
.EXTRN ANLOBS$_OBJEOMWREC
.EXTRN ANLOBS$_OBJFADPASSMECH
.EXTRN ANLOBS$_OBJGSDENV
.EXTRN ANLOBS$_OBJGSDENVFLAGS
.EXTRN ANLOBS$_OBJGSDENVPAR
.EXTRN ANLOBS$_OBJGSDPEM
.EXTRN ANLOBS$_OBJGSDPEMW
.EXTRN ANLOBS$_OBJGSDIDC
.EXTRN ANLOBS$_OBJGSDIDCENT
.EXTRN ANLOBS$_OBJGSDIDCFLAGS
.EXTRN ANLOBS$_OBJGSDIDCMATCH
.EXTRN ANLOBS$_OBJGSDIDCOBJ
.EXTRN ANLOBS$_OBJGSDIDCVLA
.EXTRN ANLOBS$_OBJGSDIDCVLB
.EXTRN ANLOBS$_OBJGSDLEPM
.EXTRN ANLOBS$_OBJGSDLPRO
.EXTRN ANLOBS$_OBJGSDLSY
.EXTRN ANLOBS$_OBJGSDPRO
.EXTRN ANLOBS$_OBJGSDPROW
.EXTRN ANLOBS$_OBJGSDPSC
.EXTRN ANLOBS$_OBJGSDPSCALIGN
.EXTRN ANLOBS$_OBJGSDPSCALLOC
.EXTRN ANLOBS$_OBJGSDPSCBASE
.EXTRN ANLOBS$_OBJGSDPSCFLAGS
.EXTRN ANLOBS$_OBJGSDREC
.EXTRN ANLOBS$_OBJGSDSPSC
.EXTRN ANLOBS$_OBJGSDSYM
.EXTRN ANLOBS$_OBJGSDSYMW
.EXTRN ANLOBS$_OBJGTXREC
.EXTRN ANLOBS$_OBJHDRIGNREC
.EXTRN ANLOBS$_OBJHEADING
.EXTRN ANLOBS$_OBJLITINDEX
.EXTRN ANLOBS$_OBJLNKREC
.EXTRN ANLOBS$_OBJLNMREC
.EXTRN ANLOBS$_OBJMHDCREATE
.EXTRN ANLOBS$_OBJMHDDNAME
.EXTRN ANLOBS$_OBJMHDPATCH
.EXTRN ANLOBS$_OBJMHDDREC
.EXTRN ANLOBS$_OBJMHDDRECSIZ
.EXTRN ANLOBS$_OBJMHDDSTRLVL
.EXTRN ANLOBS$_OBJMHDDVERSION
.EXTRN ANLOBS$_OBJMTCORRECT
.EXTRN ANLOBS$_OBJMTCINPUT
.EXTRN ANLOBS$_OBJMTCNAME
.EXTRN ANLOBS$_OBJMTCREC
.EXTRN ANLOBS$_OBJMTCSEQNUM
.EXTRN ANLOBS$_OBJMTCUIC
.EXTRN ANLOBS$_OBJMTCVERSION
.EXTRN ANLOBS$_OBJMTCWHEN
```



```
.EXTRN ANLOBS$OBJPROARGCOUNT
.EXTRN ANLOBS$OBJPROARGNUM
.EXTRN ANLOBS$OBJPSECT
.EXTRN ANLOBS$OBJSRCREC
.EXTRN ANLOBS$OBJSTATHEADING1
.EXTRN ANLOBS$OBJSTATHEADING2
.EXTRN ANLOBS$OBJSTATLINE
.EXTRN ANLOBS$OBJSTATTOTAL
.EXTRN ANLOBS$OBJSYMBOL
.EXTRN ANLOBS$OBJSYMFLAGS
.EXTRN ANLOBS$OBJTIRARGINDEX
.EXTRN ANLOBS$OBJTIRCMD
.EXTRN ANLOBS$OBJTIRCMDSTK
.EXTRN ANLOBS$OBJTBTRC
.EXTRN ANLOBS$OBJTIRREC
.EXTRN ANLOBS$OBJTIRSTOIM
.EXTRN ANLOBS$OBJTIRVIELD
.EXTRN ANLOBS$OBJTTLREC
.EXTRN ANLOBS$OBJVALUE
.EXTRN ANLOBS$OBJUVALUE
.EXTRN ANLOBS$PROTECTION
.EXTRN ANLOBS$SEVERITY
.EXTRN ANLOBS$TEXT, ANLOBS$TEXTHDR
.EXTRN ANLOBS$NOSUCHMOD
.EXTRN ANLOBS$BADDATE
.EXTRN ANLOBS$BADHDRBLKCOUNT
.EXTRN ANLOBS$BADSEVERITY
.EXTRN ANLOBS$BADSYMIST
.EXTRN ANLOBS$BADSYMCHAR
.EXTRN ANLOBS$BADSYMLN
.EXTRN ANLOBS$EXEBADFIXUPEND
.EXTRN ANLOBS$EXEBADFIXUPISD
.EXTRN ANLOBS$EXEBADFIXUPVBN
.EXTRN ANLOBS$EXEBADISDS1
.EXTRN ANLOBS$EXEBADISDTYPE
.EXTRN ANLOBS$EXEBADMATCH
.EXTRN ANLOBS$EXEBADPATCHLEN
.EXTRN ANLOBS$EXEBADOBJ
.EXTRN ANLOBS$EXEBADTYPE
.EXTRN ANLOBS$EXEBADXFERO
.EXTRN ANLOBS$EXEHDRISDLONG
.EXTRN ANLOBS$EXEHDRLONG
.EXTRN ANLOBS$EXEISDLENDZRO
.EXTRN ANLOBS$EXEISDLENGBL
.EXTRN ANLOBS$EXEISDLENPRIV
.EXTRN ANLOBS$EXENOTNATIVE
.EXTRN ANLOBS$EXTRABYTES
.EXTRN ANLOBS$FIELDFIT
.EXTRN ANLOBS$FLAGERROR
.EXTRN ANLOBS$NOTOK, ANLOBS$OBJBADIDCMATCH
.EXTRN ANLOBS$OBJBADNUM
.EXTRN ANLOBS$OBJBADPOP
.EXTRN ANLOBS$OBJBADPUSH
.EXTRN ANLOBS$OBJBADTYPE
.EXTRN ANLOBS$OBJBADVIELD
.EXTRN ANLOBS$OBJEOMBADSEV
.EXTRN ANLOBS$OBJEOMMISSING
```

```
.EXTRN ANLOBJ$_OBJFADBADA VC
.EXTRN ANLOBJ$_OBJFADBADRBC
.EXTRN ANLOBJ$_OBJGSDBADALIGN
.EXTRN ANLOBJ$_OBJGSDBADSUBTYP
.EXTRN ANLOBJ$_OBJHDRRES
.EXTRN ANLOBJ$_OBJMHDBADRECSIZ
.EXTRN ANLOBJ$_OBJMHDBADSTRLVL
.EXTRN ANLOBJ$_OBJMHDMISSING
.EXTRN ANLOBJ$_OBJNONTIRCMD
.EXTRN ANLOBJ$_OBJNOPSC
.EXTRN ANLOBJ$_OBJNULLREC
.EXTRN ANLOBJ$_OBJPOSPACE
.EXTRN ANLOBJ$_OBJPROMINMAX
.EXTRN ANLOBJ$_OBJPSCABSLEN
.EXTRN ANLOBJ$_OBJRECTOOBIG
.EXTRN ANLOBJ$_OBJTIRRES
.EXTRN ANLOBJ$_OBJUNDEFENV
.EXTRN ANLOBJ$_OBJUNDEFIT
.EXTRN ANLOBJ$_OBJUNDEFPS
.EXTRN ANALYZES_FACILITY
.EXTRN ANL$CHECK_FLAGS
.EXTRN ANL$CHECK_SYMBOL
.EXTRN ANL$CHECK_WHEN, ANL$FORMAT_ERROR
.EXTRN ANL$FORMAT_FLAGS
.EXTRN ANL$FORMAT_HEX, ANL$FORMAT_LINE
.EXTRN ANL$OBJECT_ENV_CHECK
.EXTRN ANL$OBJECT_PSECT_CHECK
.EXTRN ANL$OBJECT_PSECT_REF
.EXTRN ANL$OBJECT_RECORD_LINE
.EXTRN ANL$OBJECT_TIR_CLEAN
.EXTRN ANL$REPORT_LINE
```

```
.PSECT $CODE$,NOWRT,2
```

```
OFFC 00000
5B 00000000G 8F D0 00002
5A 0000G CF 9E 00009
59 0000G CF 9E 0000E
58 00000000G 8F D0 00013
55 08 AC D0 0001A
6C 95 0001E
03 13 00020
04 AC D5 00022
03 12 00025 1$:
01AA 31 00027 2$:
53 04 A5 9E 0002A 2$:
52 63 D0 0002E
55 DD 00031
04 AC DD 00033
57 D4 00036
03 62 91 00038
0A 12 0003B
57 D6 0003D
00000000G 8F DD 0003F
06 11 00045
00000000G 8F DD 00047 3$:

.MOVL #ANLOBJ$_OBJPSECT, R11
.MOVAB ANL$FORMAT_LINE, R10
.MOVAB ANL$FORMAT_ERROR, R9
.MOVL #ANLOBJ$_FIELDIT, R8
.MOVL THE_RECORD, R5
.TSTB (AP)
.BEQL 1$
.TSTL 4(AP)
.BNEQ 2$
.BRW 21$
.MOVAB 4(R5), R3
.MOVL (R3), SCANP
.PUSHL R5
.PUSHL RECORD_NUMBER
.CLRL R7
.CMPB (SCANP), #3
.BNEQ 3$
.INCL R7
.PUSHL #ANLOBJ$_OBJEOMREC
.BRB 4$
.PUSHL #ANLOBJ$_OBJEOMWREC
```

0572

0575
0596

0601

0602
0604
0602

0000G	CF	03	FB	0004D	4\$:	CALLS	#3, ANL\$OBJECT_RECORD_LINE	
		7E	D4	00052		CLRL	-(SP)	0605
0000G	CF	01	FB	00054		CALLS	#1, ANL\$REPORT_LINE	
54		01	90	00059		MOVB	#1, FIT_OK	0609
16		54	E9	0005C		BLBC	FIT_OK, 5\$	0611
51	02	A2	9E	0005F		MOVAB	2(R2), R1	
50		65	3C	00063		MOVZWL	(R5), R0	
50		63	C0	00066		ADDL2	(R3), R0	
50		51	D1	00069		CMPL	R1, R0	
		07	1B	0006C		BLEQU	5\$	
		58	DD	0006E		PUSHL	R8	
69		01	FB	00070		CALLS	#1, ANL\$FORMAT_ERROR	
		54	94	00073		CLRB	FIT_OK	
69		54	E9	00075	5\$:	BLBC	FIT_OK, 12\$	0612
56	01	A2	9A	00078		MOVZBL	1(SCANP), R6	0621
		56	DD	0007C		PUSHL	R6	
		08	12	0007E		BNEQ	6\$	0614
	00000000G	8F	DD	00080		PUSHL	#ANLOBJ\$_OBJEOMSEVSUC	
		3F	11	00086		BRB	11\$	
01		56	91	00088	6\$:	CPMB	R6, #1	0615
		08	12	0008B		BNEQ	7\$	
	00000000G	8F	DD	0008D		PUSHL	#ANLOBJ\$_OBJEOMSEVWRN	
		32	11	00093		BRB	11\$	
02		56	91	00095	7\$:	CPMB	R6, #2	0616
		08	12	00098		BNEQ	8\$	
	00000000G	8F	DD	0009A		PUSHL	#ANLOBJ\$_OBJEOMSEVERR	
		25	11	000A0		BRB	11\$	
03		56	91	000A2	8\$:	CPMB	R6, #3	0617
		08	12	000A5		BNEQ	9\$	
	00000000G	8F	DD	000A7		PUSHL	#ANLOBJ\$_OBJEOMSEVABT	
		18	11	000AD		BRB	11\$	
04		56	91	000AF	9\$:	CPMB	R6, #4	0618
		0D	1F	000B2		BLSSU	10\$	
0A		56	91	000B4		CPMB	R6, #10	
		08	1A	000B7		BGTRU	10\$	
	00000000G	8F	DD	000B9		PUSHL	#ANLOBJ\$_OBJEOMSEVRER	
		06	11	000BF		BRB	11\$	
	00000000G	8F	DD	000C1	10\$:	PUSHL	#ANLOBJ\$_OBJEOMSEVIGN	0619
		01	DD	000C7	11\$:	PUSHL	#1	0613
		7E	D4	000C9		CLRL	-(SP)	
6A		04	FB	000CB		CALLS	#4, ANL\$FORMAT_LINE	
04		56	91	000CE		CPMB	R6, #4	0622
		0E	1F	000D1		BLSSU	12\$	
0A		56	91	000D3		CPMB	R6, #10	
		09	1A	000D6		BGTRU	12\$	
	00000000G	8F	DD	000D8		PUSHL	#ANLOBJ\$_OBJEOMBADSEV	0623
69		01	FB	000DE		CALLS	#1, ANL\$FORMAT_ERROR	
02		65	B1	000E1	12\$:	CPW	(R5), #2	0628
		03	1A	000E4		BGTRU	13\$	
	00EB	31	000E6		BRW	21\$		
37		57	E9	000E9	13\$:	BLBC	R7, 15\$	0634
16		54	E9	000EC		BLBC	FIT_OK, 14\$	0635
51	03	A2	9E	000EF		MOVAB	3(R2), R1	
50		65	3C	000F3		MOVZWL	(R5), R0	
50		63	C0	000F6		ADDL2	(R3), R0	
50		51	D1	000F9		CMPL	R1, R0	
		07	1B	000FC		BLEQU	14\$	

		58	DD	000FE	PUSHL	R8		
69		01	FB	00100	CALLS	#1, ANL\$FORMAT_ERROR		
		54	94	00103	CLRB	FIT_OK		
69		54	E9	00105	BLBC	FIT_OK, 18\$	14\$:	0636
7E	02	A2	9A	00108	MOVZBL	2(SCANP), -(SP)		0637
		58	DD	0010C	PUSHL	R11		
		01	DD	0010E	PUSHL	#1		
		7E	D4	00110	CLRL	-(SP)		
6A		04	FB	00112	CALLS	#4, ANL\$FORMAT_LINE		
7E	02	A2	9A	00115	MOVZBL	2(SCANP), -(SP)		0638
0000G		01	FB	00119	CALLS	#1, ANL\$OBJECT_PSECT_REF		
52		03	C0	0011E	ADDL2	#3, SCANP		0639
		35	11	00121	BRB	17\$		0634
72		54	E9	00123	BLBC	FIT_OK, 20\$	15\$:	0644
51	04	A2	9E	00126	MOVAB	4(R2), R1		
50		65	3C	0012A	MOVZWL	(R5), R0		
50		63	C0	0012D	ADDL2	(R3), R0		
50		51	D1	00130	CMPL	R1, R0		
		07	1B	00133	BLEQU	16\$		
		58	DD	00135	PUSHL	R8		
69		01	FB	00137	CALLS	#1, ANL\$FORMAT_ERROR		
		54	94	0013A	CLRB	FIT_OK		
59		54	E9	0013C	BLBC	FIT_OK, 20\$	16\$:	0645
7E	02	A2	3C	0013F	MOVZWL	2(SCANP), -(SF)		0646
		58	DD	00143	PUSHL	R11		
		01	DD	00145	PUSHL	#1		
		7E	D4	00147	CLRL	-(SP)		
6A		04	FB	00149	CALLS	#4, ANL\$FORMAT_LINE		
7E	02	A2	3C	0014C	MOVZWL	2(SCANP), -(SP)		0647
0000G		01	FB	00150	CALLS	#1, ANL\$OBJECT_PSECT_REF		
52		04	C0	00155	ADDL2	#4, SCANP		0648
3D		54	E9	00158	BLBC	FIT_OK, 20\$	17\$:	0654
51	04	A2	9E	0015B	MOVAB	4(R2), R1		
50		65	3C	0015F	MOVZWL	(R5), R0		
50		63	C0	00162	ADDL2	(R3), R0		
50		51	D1	00165	CMPL	R1, R0		
		07	1B	00168	BLEQU	18\$		
		58	DD	0016A	PUSHL	R8		
69		01	FB	0016C	CALLS	#1, ANL\$FORMAT_ERROR		
		54	94	0016F	CLRB	FIT_OK		
24		54	E9	00171	BLBC	FIT_OK, 20\$	18\$:	0655
	00000000G	62	DD	00174	PUSHL	(SCANP)		0656
		8F	DD	00176	PUSHL	#ANLOBJ\$_OBJVALUE		
		01	DD	0017C	PUSHL	#1		
		7E	D4	0017E	CLRL	-(SP)		
6A		04	FB	00180	CALLS	#4, ANL\$FORMAT_LINE		
3FFFFFFF		62	D1	00183	CMPL	(SCANP), #1073741823		0657
		09	1B	0018A	BLEQU	19\$		
	00000000G	8F	DD	0018C	PUSHL	#ANLOBJ\$_OBJPOSPACE		0658
69		01	FB	00192	CALLS	#1, ANL\$FORMAT_ERROR		
52		04	C0	00195	ADDL2	#4, SCANP	19\$:	0659
50		65	3C	00198	MOVZWL	(R5), R0	20\$:	0664
63		50	C1	0019B	ADDL3	R0, (R3), R3		
52		53	D1	0019F	CMPL	R3, SCANP		
		30	1B	001A2	BLEQU	21\$		
	0000'	CF	9F	001A4	PUSHAB	TRANSFER_FLAGS_DEF		0669
7E		62	9A	001A8	MOVZBL	(SCANP), -(SP)		

OBJMISC
V04-000

OBJMISC - Analyze Miscellaneous Object Records
ANL\$OBJECT_EOM - Analyze EOM and EOMW Records

D 2
15-Sep-1984 23:42:42
14-Sep-1984 11:52:57

VAX-11 Bliss-32 V4.0-742
[ANALYZ.SRC]OBJMISC.B32;1

Page 14
(4)

		00000000G	8F	DD	001AB	PUSHL	#ANLOBJ\$OBJEOMFLAGS	
			01	DD	001B1	PUSHL	#1	
0000G	CF		04	FB	001B3	CALLS	#4, ANL\$FORMAT_FLAGS	
		0000'	CF	9F	001B8	PUSHAB	TRANSFER_FLAGS_DEF	0670
	7E		62	9A	001BC	MOVZBL	(SCANP), -(SP)	
0000G	CF		02	FB	001BF	CALLS	#2, ANL\$CHECK_FLAGS	
			52	D6	001C4	INCL	SCANP	0671
	52		53	D1	001C6	CMPL	R3, SCANP	0675
			09	1B	001C9	BLEQU	21\$	
		00000000G	8F	DD	001CB	PUSHL	#ANLOBJ\$EXTRABYTES	0676
	69		01	FB	001D1	CALLS	#1, ANL\$FORMAT_ERROR	
0000G	CF		00	FB	001D4	CALLS	#0, ANL\$OBJECT_TIR_CLEAN	0685
0000G	CF		00	FB	001D9	CALLS	#0, ANL\$OBJECT_PSETT_CHECK	0690
0000G	CF		00	FB	001DE	CALLS	#0, ANL\$OBJECT_ENV_CHECK	0691
0000'	CF	0800	8F	3C	001E3	MOVZWL	#2048, MMD_RECORD_SIZE	0695
			04	001EA	RET			0699

: Routine Size: 491 bytes, Routine Base: \$CODE\$ + 0000

: 268 0700 1

```
270 0701 1 %sbttl 'ANL$OBJECT_HDR - Analyze Object Header Records'
271 0702 1 **
272 0703 1 Functional Description:
273 0704 1 This routine is called to analyze header records from object files.
274 0705 1
275 0706 1 Formal Parameters:
276 0707 1 record_number The record number of this header record.
277 0708 1 the_record The address of the descriptor of this record.
278 0709 1
279 0710 1 Implicit Inputs:
280 0711 1 global data
281 0712 1
282 0713 1 Implicit Outputs:
283 0714 1 global data
284 0715 1
285 0716 1 Returned Value:
286 0717 1 none
287 0718 1
288 0719 1 Side Effects:
289 0720 1
290 0721 1 --
291 0722 1
292 0723 1
293 0724 2 global routine anl$object_hdr(record_number,the_record): novalue = begin
294 0725 2
295 0726 2 bind
296 0727 2 record_dsc = .the_record: descriptor;
297 0728 2
298 0729 2 local
299 0730 2 status: long,
300 0731 2 scanp: ref block[.byte],
301 0732 2 fit_ok: byte;
302 0733 2
303 0734 2
304 0735 2 ! Decide what to do based on the header type. If there isn't one, forget it.
305 0736 2
306 0737 2 scanp = .record_dsc[ptr];
307 0738 2 fit_ok = true;
308 0739 2 ensure_field_fit(obj$b_subtyp,record_dsc);
309 0740 2 if not .fit_ok then
310 0741 2 return;
311 0742 2
312 0743 2 selectoneu .scanp[obj$b_subtyp] of set
313 0744 2 [obj$c_hdr_mhd]: anl$object_hdr_mhd(.record_number,record_dsc);
314 0745 2
315 0746 2 [obj$c_hdr_lnm,
316 0747 2 obj$c_hdr_src,
317 0748 2 obj$c_hdr_ttl,
318 0749 2 obj$c_hdr_cpr,
319 0750 2 obj$c_hdr_gtx]: anl$object_hdr_text(.record_number,record_dsc);
320 0751 2
321 0752 2 [obj$c_hdr_mtc]: anl$object_hdr_mtc(.record_number,record_dsc);
322 0753 2
323 0754 2 [mhd$c_maxhdrtyp+1
324 0755 2 to 100]: (anl$format_error(anlobj$_objhdrres,.record_number,.scanp[obj$b_subtyp]));
325 0756 2 anl$report_line(0);
326 0757 2 anl$format_hex(1,record_dsc););
```


OBJMISC
V04-000OBJMISC - Analyze Miscellaneous Object Records F 2
ANLSOBJECT_HDR - Analyze Object Header Records 15-Sep-1984 23:42:42 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 11:52:57 [ANALYZ.SRC]OBJMISC.B32:1Page 16
(5)327
328
329
330
331
332
333
334
335
336

```
0758 2  
0759 [101 to 255]: (anl$format_line(0,0,anlobj$objhdrignrec,.record_number,  
0760 scanp[obj$b_subtyp],.record_dsc[ten]);  
0761 anl$report_line(0);  
0762 anl$format_hex(1,record_dsc););  
0763 tes;  
0764  
0765 return;  
0766  
0767 1 end;
```

```
001C 00000  
53 08 AC D0 00002  
52 04 A3 D0 00006  
54 01 90 0000A  
1D 54 E9 0000D  
51 02 A2 9E 00010  
50 63 3C 00014  
50 04 A3 C0 00017  
50 51 D1 0001B  
0D 1B 0001E  
0000G CF 00000000G 8F DD 00020  
01 01 FB 00026  
54 54 94 0002B  
7E 54 E9 0002D 1$:  
52 01 A2 9A 00030  
0B 12 00034  
53 DD 00036  
04 AC DD 00038  
0000V CF 02 FB 0003B  
04 04 00040  
04 52 91 00041 2$:  
06 05 1B 00044  
52 91 00046  
0B 12 00049  
53 DD 0004B 3$:  
04 AC DD 0004D  
0000V CF 02 FB 00050  
04 04 00055  
05 52 91 00056 4$:  
0B 12 00059  
53 DD 0005B  
04 AC DD 0005D  
0000V CF 02 FB 00060  
04 04 00065  
07 52 91 00066 5$:  
1B 1F 00069  
64 8F 52 91 0006B  
12 1A 0006F  
52 DD 00071  
04 AC DD 00073  
0000G CF 00000000G 8F DD 00076  
03 FB 0007C
```

```
.ENTRY ANLSOBJECT_HDR, Save R2,R3,R4  
MOVL THE_RECORD, R3  
MOVL 4(R3), SCANP  
MOVB #1, FIT_OK  
BLBC FIT_OK, 1$  
MOVAB 2(R2), R1  
MOVZWL (R3), R0  
ADDL2 4(R3), R0  
CML R1, R0  
BLEQU 1$  
PUSHL #ANLOBJ$ FIELDFIT  
CALLS #1, ANLSFORMAT_ERROR  
CLRB FIT_OK  
BLBC FIT_OK, 8$  
MOVZBL 1(SCANP), R2  
BNEQ 2$  
PUSHL R3  
PUSHL RECORD_NUMBER  
CALLS #2, ANLSOBJECT_HDR_MHD  
RET  
CMPB R2, #4  
BLEQU 3$  
CMPB R2, #6  
BNEQ 4$  
PUSHL R3  
PUSHL RECORD_NUMBER  
CALLS #2, ANLSOBJECT_HDR_TEXT  
RET  
CMPB R2, #5  
BNEQ 5$  
PUSHL R3  
PUSHL RECORD_NUMBER  
CALLS #2, ANLSOBJECT_HDR_MTC  
RET  
CMPB R2, #7  
BLSSU 6$  
CMPB R2, #100  
BGTRU 6$  
PUSHL R2  
PUSHL RECORD_NUMBER  
PUSHL #ANLOBJ$ OBJHDRRES  
CALLS #3, ANLSFORMAT_ERROR
```

0724
0727
0737
0738
07390740
0743
0744

0746

0750

0752

0754

0755

4
5)

OBJMISC
V04-000

OBJMISC - Analyze Miscellaneous Object Records 15-Sep-1984 23:42:42
ANLSOBJECT_HDR - Analyze Object Header Records 14-Sep-1984 11:52:57

VAX-11 Bliss-32 V4.0-742
[ANALYZ.SRC]OBJMISC.B32;1

Page 17
(5)

65	8F	1B	11	00081	BRB	7\$...	0756
		52	91	00083	6\$: CMPB	R2, #101	...	0759
	7E	25	1F	00087	BLSSU	8\$...	
		63	3C	00089	MOVZWL	(R3), -(SP)	...	0760
		52	DD	0008C	PUSHL	R2	...	
		AC	DD	0008E	PUSHL	RECORD NUMBER	...	0759
		8F	DD	00091	PUSHL	#ANL\$OBJ\$_OBJHDRIGNREC	...	
		7E	7C	00097	CLRG	-(SP)	...	
0000G	CF	06	FB	00099	CALLS	#6, ANLSFORMAT_LINE	...	
		7E	D4	0009E	7\$: CLRL	-(SP)	...	0761
0000G	CF	01	FB	000A0	CALLS	#1, ANLSREPORT_LINE	...	
		53	DD	000A5	PUSHL	R3	...	0762
		01	DD	000A7	PUSHL	#1	...	
0000G	CF	02	FB	000A9	CALLS	#2, ANLSFORMAT_HEX	...	
		04	000AE	8\$: RET			...	0767

; Routine Size: 175 bytes, Routine Base: \$CODE\$ + 01EB

OBJMISC
V04-000OBJMISC - Analyze Miscellaneous Object Records H 2
ANL\$OBJECT_HDR_MHD - Analyze Module Header Reco 15-Sep-1984 23:42:42
14-Sep-1984 11:52:57VAX-11 Bliss-32 V4.0-742
[ANALYZ.SRC]OBJMISC.B32;1Page 18
(6)

```
338 0768 1 %sbtcl 'ANL$OBJECT_HDR_MHD - Analyze Module Header Record'
339 0769 1 **
340 0770 1 Functional Description:
341 0771 1 This routine is called to analyze the module header record.
342 0772 1
343 0773 1 Formal Parameters:
344 0774 1 record_number The number of this record in the object file.
345 0775 1 the_record The address of the descriptor of the record.
346 0776 1
347 0777 1 Implicit Inputs:
348 0778 1 global data
349 0779 1
350 0780 1 Implicit Outputs:
351 0781 1 global data
352 0782 1
353 0783 1 Returned Value:
354 0784 1 none
355 0785 1
356 0786 1 Side Effects:
357 0787 1
358 0788 1 --
359 0789 1
360 0790 1
361 0791 2 global routine anl$object_hdr_mhd(record_number,the_record): novalue = begin
362 0792 2
363 0793 2 bind
364 0794 2 record_dsc = .the_record: descriptor;
365 0795 2
366 0796 2 local
367 0797 2 status: long,
368 0798 2 scanp: ref block[,byte],
369 0799 2 fit_ok: byte,
370 0800 2 work_dsc: descriptor;
371 0801 2
372 0802 2
373 0803 2 ! We begin by printing a record line for this module header.
374 0804 2
375 0805 2 anl$object_record_line(anlobj$_objmhdrec,.record_number,record_dsc);
376 0806 2 anl$report_line(0);
377 0807 2
378 0808 2 ! Now we print the structure level and make sure it is valid.
379 0809 2
380 0810 2 scanp = .record_dsc[ptr];
381 0811 2 fit_ok = true;
382 0812 2 ensure_field_fit(mhd$b_strlvl,record_dsc);
383 0813 2 if .fit_ok then (
384 0814 2 anl$format_line(0,1,anlobj$_objmhdstrlvl,.scanp[mhd$b_strlvl]);
385 0815 2 if .scanp[mhd$b_strlvl] gtru obj$c_strlvl then
386 0816 2 anl$format_error(anlobj$_objmhdbadstrlvl,obj$c_strlvl);
387 0817 2 );
388 0818 2
389 0819 2 ! Now we print the maximum record size and make sure it's valid. We also
390 0820 2 ! save it for future use.
391 0821 2
392 0822 2 ensure_field_fit(mhd$w_recsiz,record_dsc);
393 0823 2 if .fit_ok then (
394 0824 2 anl$format_line(0,1,anlobj$_objmhdrecsiz,.scanp[mhd$w_recsiz]);
```


OBJMISC
V04-000OBJMISC - Analyze Miscellaneous Object Records 15-Sep-1984 23:42:42
ANLSOBJECT_HDR_MHD - Analyze Module Header Reco 14-Sep-1984 11:52:57
YAX-11 Bliss-32 V4.0-742
[ANALYZ.SRC]OBJMISC.B32;1Page 19
(6)

```
395 0825 3      if .scanp[mhd$w_recsize] gtru obj$c_maxrecsize then
396 0826      anl$format_error(anlobj$_objmhdbadrecsize,obj$c_maxrecsize);
397 0827      mhd_record_size = .scanp[mhd$w_recsize];
398 0828  );
399 0829
400 0830  ! Now we print the module name and make sure it's valid.
401 0831
402 0832  ensure_asis_fit(mhd$b_namlen,record_dsc,work_dsc);
403 0833  if .fit_ok then (
404 0834      anl$format_line(0,1,anlobj$_objmhdname,.work_dsc[len],.work_dsc[ptr]);
405 0835      anl$check_symbol(work_dsc,$h$c_maxnamlen);
406 0836      scanp = .work_dsc[ptr] + .work_dsc[len];
407 0837  );
408 0838
409 0839  ! Now we print the module version and make sure it's valid.
410 0840
411 0841  ensure_asis_fit(0,0,8,0,record_dsc,work_dsc);
412 0842  if .fit_ok then (
413 0843      anl$format_line(0,1,anlobj$_objmhdversion,.work_dsc[len],.work_dsc[ptr]);
414 0844      if (.work_dsc[len] lssu 1) or (.work_dsc[len] gtru obj$c_symsize) then
415 0845          anl$format_error(anlobj$_badsymlen,obj$c_symsize);
416 0846      scanp = .work_dsc[ptr] + .work_dsc[len];
417 0847  );
418 0848
419 0849  ! Now we print the creation date/time and make sure it's valid.
420 0850
421 0851  ensure_field_fit(0,0,17*8,0,record_dsc);
422 0852  if .fit_ok then (
423 0853      build_descriptor(work_dsc,17,.scanp);
424 0854      anl$format_line(0,1,anlobj$_objmhdcreate,work_dsc);
425 0855      anl$check_when(work_dsc);
426 0856      scanp = .scanp + 17;
427 0857  );
428 0858
429 0859  ! If we're at the end of the record, no problem, just return
430 0860
431 0861  if .record_dsc[ptr] + .record_dsc[len] gequ .scanp then
432 0862      return;
433 0863
434 0864  ! If there is a last patch date/time field, print it and make sure
435 0865  ! it's valid. It can be blank, full of nulls or contain a date.
436 0866
437 0867  ensure_field_fit(0,0,17*8,0,record_dsc);
438 0868  if .fit_ok then (
439 0869      build_descriptor(work_dsc,17,.scanp);
440 0870      if not (ch$neq(17,.scanp,0,0,0)) then ! if nothing but 0's, fill with blanks
441 0871          ch$copy(0,0,17,.work_dsc[ptr],0);
442 0872      anl$format_line(0,1,anlobj$_objmhdpatch,work_dsc);
443 0873      if ch$neq(17,.work_dsc[ptr],0,0,0) then
444 0874          anl$check_when(work_dsc);
445 0875      scanp = .scanp + 17;
446 0876  );
447 0877
448 0878  ! Finally, we ensure that there are no spurious bytes at the end.
449 0879
450 0880  if .record_dsc[ptr]+.record_dsc[len] gtru .scanp then
451 0881      anl$format_error(anlobj$_extrabytes);
```

OBJMISC
V04-000OBJMISC - Analyze Miscellaneous Object Records 15-Sep-1984 23:42:42
ANLSOBJECT_HDR_MHD - Analyze Module Header Reco 14-Sep-1984 11:52:57VAX-11 Bliss-32 V4.0-742
[ANALYZ.SRC]OBJMISC.B32:1Page 20
(6): 452
: 453
: 454
: 4550882 2
0883 2 return;
0884 2
0885 1 end;

				07FC 00000	.ENTRY	ANLSOBJECT_HDR_MHD, Save R2,R3,R4,R5,R6,R7,-	0791
5A	0000G	CF	9E	00002	MOVAB	R8,R9,R10	
59	0000G	CF	9E	00007	MOVAB	ANLSFORMAT_LINE, R10	
58	00000000G	8F	DD	0000C	MOVL	ANLSFORMAT_ERROR, R9	
5E		08	C2	00013	SUBL2	#ANLOBS_FIELDFIT, R8	
54	08	AC	DD	00016	MOVL	#8, SP	
		54	DD	0001A	MOVL	THE_RECORD, R4	0794
					PUSHL	R4	0805
	04	AC	DD	0001C	PUSHL	RECORD_NUMBER	
	00000000G	8F	DD	0001F	PUSHL	#ANLOBS_OBJMHDREC	
0000G	CF	03	FB	00025	CALLS	#3, ANLSOBJECT_RECORD_LINE	
		7E	D4	0002A	CLRL	-(SP)	0806
0000G	CF	01	FB	0002C	CALLS	#1, ANLSREPORT_LINE	
	04	A4	DD	00031	MOVL	4(R4), R2	0810
52		52	DD	00035	MOVL	R2, SCANP	
56		01	90	00038	MOVB	#1, FIT_OK	0811
53		53	E9	0003B	BLBC	FIT_OK, 3\$	0812
53					MOVAB	3(R8), R1	
51	03	A6	9E	0003E	MOVZWL	(R4), R0	
50		64	3C	00042	ADDL2	R2, R0	
50		52	C0	00045	CMPL	R1, R0	
50		51	D1	00048	BLEQU	1\$	
		07	1B	0004B	PUSHL	R8	
		58	DD	0004D	CALLS	#1, ANLSFORMAT_ERROR	
69		01	FB	0004F	CLRB	FIT_OK	
		53	94	00052	BLBC	FIT_OK, 5\$	0813
6A		53	E9	00054	MOVZBL	2(SCANP), -(SP)	0814
7E	02	A6	9A	00057	PUSHL	#ANLOBS_OBJMHDSTRLVL	
	00000000G	8F	DD	0005B	PUSHL	#1	
		01	DD	00061	CLRL	-(SP)	
		7E	D4	00063	CALLS	#4, ANLSFORMAT_LINE	
6A		04	FB	00065	TSTB	2(SCANP)	0815
	02	A6	95	00068	BEQL	2\$	
		08	13	0006B	CLRL	-(SP)	0816
		7E	D4	0006D	PUSHL	#ANLOBS_OBJMHDBADSTRLVL	
	00000000G	8F	DD	0006F	CALLS	#2, ANLSFORMAT_ERROR	
69		02	FB	00075	BLBC	FIT_OK, 7\$	0822
68		53	E9	00078	MOVB	5(R8), R1	
51	05	A6	9E	0007B	MOVZWL	(R4), R0	
50		64	3C	0007F	ADDL2	R2, R0	
50		52	C0	00082	CMPL	R1, R0	
50		51	D1	00085	BLEQU	3\$	
		07	1B	00088	PUSHL	R8	
		58	DD	0008A	CALLS	#1, ANLSFORMAT_ERROR	
69		01	FB	0008C	CLRB	FIT_OK	
		53	94	0008F	BLBC	FIT_OK, 8\$	0823
72		53	E9	00091	MOVZWL	3(SCANP), -(SP)	0824
7E	03	A6	3C	00094	PUSHL	#ANLOBS_OBJMHDRECSIZ	
	00000000G	8F	DD	00098			

			01	DD	0009E	PUSHL	#1		
			7E	D4	000A0	CLRL	-(SP)		
0800	6A		04	FB	000A2	CALLS	#4, ANLSFORMAT_LINE		
	8F	03	A6	B1	000A5	CMPW	3(SCANP), #2048		0825
			0E	1B	000AB	BLEQU	4\$		
	7E	0800	8F	3C	000AD	MOVZWL	#2048, -(SP)		0826
		00000000G	8F	DD	000B2	PUSHL	#ANL\$OBJ\$OBJMHDBADRECSIZ		
	69		02	FB	000B8	CALLS	#2, ANLSFORMAT_ERROR		
0000'	CF	03	A6	3C	000BB	MOVZWL	3(SCANP), MHD_RECORD_SIZE		0827
	6A		53	E9	000C1	BLBC	FIT_OK, 9\$		0832
	51	06	A6	9E	000C4	MOVAB	6(R6), R1		
	50		64	3C	000C8	MOVZWL	(R4), R0		
	50		52	C0	000CB	ADDL2	R2, R0		
	50		51	D1	000CE	CMPL	R1, R0		
			07	1B	000D1	BLEQU	6\$		
			58	DD	000D3	PUSHL	R8		
	69		01	FB	000D5	CALLS	#1, ANLSFORMAT_ERROR		
			53	94	000D8	CLRB	FIT_OK		
	75		53	E9	000DA	BLBC	FIT_OK, 11\$		
04	6E	05	A6	9A	000DD	MOVZBL	5(SCANP), WORK_DSC		
	AE	06	A6	9E	000E1	MOVAB	6(R6), WORK_DSC+4		
	69		53	E9	000E6	BLBC	FIT_OK, 11\$		
	50		6E	3C	000E9	MOVZWL	WORK_DSC, R0		
	50		08	C6	000EC	DIVL2	#8, R0		
	51	01	A046	9E	000EF	MOVAB	1(R0)[SCANP], R1		
	50		64	3C	000F4	MOVZWL	(R4), R0		
	50		52	C0	000F7	ADDL2	R2, R0		
	50		51	D1	000FA	CMPL	R1, R0		
			07	1B	000FD	BLEQU	8\$		
			58	DD	000FF	PUSHL	R8		
	69		01	FB	00101	CALLS	#1, ANLSFORMAT_ERROR		
			53	94	00104	CLRB	FIT_OK		
	69		53	E9	00106	BLBC	FIT_OK, 12\$		0833
		04	AE	DD	00109	PUSHL	WORK_DSC+4		0834
	7E	04	AE	3C	0010C	MOVZWL	WORK_DSC, -(SP)		
		00000000G	8F	DD	00110	PUSHL	#ANL\$OBJ\$OBJMHDDNAME		
			01	DD	00116	PUSHL	#1		
			7E	D4	00118	CLRL	-(SP)		
	6A		05	FB	0011A	CALLS	#5, ANLSFORMAT_LINE		
			27	DD	0011D	PUSHL	#39		0835
		04	AE	9F	0011F	PUSHAB	WORK_DSC		
0000G	CF		02	FB	00122	CALLS	#2, ANLSCHECK_SYMBOL		
	56		6E	3C	00127	MOVZWL	WORK_DSC, SCANP		0836
	56	04	AE	C0	0012A	ADDL2	WORK_DSC+4, SCANP		
	73		53	E9	0012E	BLBC	FIT_OK, 15\$		0841
	51	01	A6	9E	00131	MOVAB	1(R6), R1		
	50		64	3C	00135	MOVZWL	(R4), R0		
	50		52	C0	00138	ADDL2	R2, R0		
	50		51	D1	0013B	CMPL	R1, R0		
			07	1B	0013E	BLEQU	10\$		
			58	DD	00140	PUSHL	R8		
	69		01	FB	00142	CALLS	#1, ANLSFORMAT_ERROR		
			53	94	00145	CLRB	FIT_OK		
	73		53	E9	00147	BLBC	FIT_OK, 16\$		
	6E		66	9A	0014A	MOVZBL	(SCANP), WORK_DSC		
04	AE	01	A6	9E	0014D	MOVAB	1(R6), WORK_DSC+4		
	68		53	E9	00152	BLBC	FIT_OK, 16\$		

50		6E	3C	00155	MOVZWL	WORK_DSC, R0	
50		08	C6	00158	DIVL2	#8, R0	
51	01	A046	9E	00158	MOVAB	1(R0)[SCANP], R1	
50		64	3C	00160	MOVZWL	(R4), R0	
50		52	C0	00163	ADDL2	R2, R0	
50		51	D1	00166	CMPL	R1, R0	
		07	1B	00169	BLEQU	12\$	
		58	DD	00168	PUSHL	R8	
69		01	FB	0016D	CALLS	#1, ANL\$FORMAT_ERROR	
		53	94	00170	CLRB	FIT_OK	
6B		53	E9	00172	BLBC	FIT_OK, 17\$	0842
	04	AE	DD	00175	PUSHL	WORK_DSC+4	0843
7E	04	AE	3C	00178	MOVZWL	WORK_DSC, -(SP)	
	00000000G	8F	DD	0017C	PUSHL	#ANL\$OBJ\$_OBJMHDVERSION	
		01	DD	00182	PUSHL	#1	
		7E	D4	00184	CLRL	-(SP)	
6A		05	FB	00186	CALLS	#5, ANL\$FORMAT_LINE	
		6E	B5	00189	TSTW	WORK_DSC	0844
		05	13	0018B	BEQL	13\$	
1F		6E	B1	0018D	CMPL	WORK_DSC, #31	
		0B	1B	00190	BLEQU	14\$	
		1F	DD	00192	PUSHL	#31	0845
	00000000G	8F	DD	00194	PUSHL	#ANL\$OBJ\$_BADSYMLEN	
69		02	FB	0019A	CALLS	#2, ANL\$FORMAT_ERROR	
56		6E	3C	0019D	MOVZWL	WORK_DSC, SCANP	0846
56	04	AE	C0	001A0	ADDL2	WORK_DSC+4, SCANP	
39		53	E9	001A4	BLBC	FIT_OK, 17\$	0851
51	11	A6	9E	001A7	MOVAB	17(R6), R1	
50		64	3C	001AB	MOVZWL	(R4), R0	
50		52	C0	001AE	ADDL2	R2, R0	
50		51	D1	001B1	CMPL	R1, R0	
		07	1B	001B4	BLEQU	16\$	
		58	DD	001B6	PUSHL	R8	
69		01	FB	001B8	CALLS	#1, ANL\$FORMAT_ERROR	
		53	94	001BB	CLRB	FIT_OK	
20		53	E9	001BD	BLBC	FIT_OK, 17\$	0852
6E		11	D0	001C0	MOVL	#17, WORK_DSC	0853
04	AE	56	D0	001C3	MOVL	SCANP, WORK_DSC+4	
		5E	DD	001C7	PUSHL	SP	0854
	00000000G	8F	DD	001C9	PUSHL	#ANL\$OBJ\$_OBJMHDCREATE	
		01	DD	001CF	PUSHL	#1	
		7E	D4	001D1	CLRL	-(SP)	
6A		04	FB	001D3	CALLS	#4, ANL\$FORMAT_LINE	
		5E	DD	001D6	PUSHL	SP	0855
0000G	CF	01	FB	001D8	CALLS	#1, ANL\$CHECK_WHEN	
56		11	C0	001DD	ADDL2	#17, SCANP	0856
57		64	3C	001E0	MOVZWL	(R4), R7	0861
57		52	C0	001E3	ADDL2	R2, R7	
56		57	D1	001E6	CMPL	R7, SCANP	
		6C	1E	001E9	BGEQU	22\$	
58		53	E9	001EB	BLBC	FIT_OK, 21\$	0867
50	11	A6	9E	001EE	MOVAB	17(R6), R0	
57		50	D1	001F2	CMPL	R0, R7	
		07	1B	001F5	BLEQU	18\$	
		58	DD	001F7	PUSHL	R8	
69		01	FB	001F9	CALLS	#1, ANL\$FORMAT_ERROR	
		53	94	001FC	CLRB	FIT_OK	

OBJMISC
V04-000OBJMISC - Analyze Miscellaneous Object Records 15-Sep-1984 23:42:42
ANL\$OBJECT_HDR_MHD - Analyze Module Header Reco 14-Sep-1984 11:52:57VAX-11 Bliss-32 V4.0-742
[ANALYZ.SRC]OBJMISC.B32;1Page 23
(6)

			48		53	E9	001FE	18\$:	BLBC	FIT OK, 21\$	0868
			6E		11	D0	00201		MOVL	#17, WORK_DSC	0869
		04	AE		56	D0	00204		MOVL	SCANP, WORK_DSC+4	
00	00		66		11	2D	00208		CMPC5	#17, (SCANPT, #0, #0, @#^X00000000	0870
				00000000	9F		0020D				
					0F	12	00212		BNEQ	19\$	
04	AE				00	2C	00214		MOVCS	#0, @#^X00000000, #17, WORK_DSC+4, -	0871
		11	00000000	9F	9F		0021E			@#^X00000020	
				00000020	5E	DD	00223	19\$:	PUSHL	SP	0872
				00000000G	8F	DD	00225		PUSHL	#ANLOBJ\$_OBJMHDPATCH	
					01	DD	0022B		PUSHL	#1	
					7E	D4	0022D		CLRL	-(SP)	
			6A		04	FB	0022F		CALLS	#4, ANL\$FORMAT_LINE	
00	20	04	BE		11	2D	00232		CMPC5	#17, @WORK_DSC+4, #32, #0, @#^X00000000	0873
				00000000	9F		00238				
					07	13	0023D		BEQL	20\$	
					5E	DD	0023F		PUSHL	SP	0874
		0000G	CF		01	FB	00241		CALLS	#1, ANL\$CHECK_WHEN	
			56		11	C0	00246	20\$:	ADDL2	#17, SCANP	0875
			56		57	D1	00249	21\$:	CMPL	R7, SCANP	0880
					09	1B	0024C		BLEQU	22\$	
				00000000G	8F	DD	0024E		PUSHL	#ANLOBJ\$_EXTRABYTES	0881
			69		01	FB	00254		CALLS	#1, ANL\$FORMAT_ERROR	
					04		00257	22\$:	RET		0885

; Routine Size: 600 bytes, Routine Base: \$CODE\$ + 029A

```

: 457      0886 1 %sbttl 'ANL$OBJECT_RECORD_SIZE - Check Object Record Size'
: 458      0887 1 ++
: 459      0888 1 Functional Description:
: 460      0889 1 This little routine is called to check the size of an object record
: 461      0890 1 against the maximum size specified in the module header. We assume
: 462      0891 1 the maximum size has been retrieved by now.
: 463      0892 1
: 464      0893 1 Formal Parameters:
: 465      0894 1 size Size of the object record to check.
: 466      0895 1
: 467      0896 1 Implicit Inputs:
: 468      0897 1 global data
: 469      0898 1
: 470      0899 1 Implicit Outputs:
: 471      0900 1 global data
: 472      0901 1
: 473      0902 1 Returned Value:
: 474      0903 1 none
: 475      0904 1
: 476      0905 1 Side Effects:
: 477      0906 1
: 478      0907 1 --
: 479      0908 1
: 480      0909 1
: 481      0910 2 global routine anl$object_record_size(size): novalue = begin
: 482      0911 2
: 483      0912 2
: 484      0913 2 ! Just check the size and print an error message if too large.
: 485      0914 2
: 486      0915 2 if .size gtru .mhd_record_size then
: 487      0916 2 anl$format_error(anl$obj$_objrectoobig,.mhd_record_size);
: 488      0917 2
: 489      0918 2 return;
: 490      0919 2
: 491      0920 1 end;
```

```

0000' CF 04 AC D1 00002 .ENTRY ANL$OBJECT_RECORD_SIZE, Save nothing : 0910
OF 1B 00008 CMPL SIZE, MHD_RECORD_SIZE : 0915
0000' CF DD 0000A BLEQU 1$ :
0000G CF 00000000G BF DD 0000E PUSHL MHD_RECORD_SIZE : 0916
02 FB 00014 PUSHL #ANL$OBJ$_OBJRECTOOBIG
04 00019 1$ CALLS #2, ANL$FORMAT_ERROR : 0920
RET
```

; Routine Size: 26 bytes, Routine Base: \$CODE\$ + 04F2


```
493 0921 1 %sbttl 'ANL$OBJECT_HDR_TEXT - Analyze Text Header Records'
494 0922 1 **
495 0923 1 Functional Description:
496 0924 1 This routine is called to analyze the header records that just
497 0925 1 contain text.
498 0926 1
499 0927 1 Formal Parameters:
500 0928 1 record_number Number of this object record.
501 0929 1 the_record Address of a descriptor of the record.
502 0930 1
503 0931 1 Implicit Inputs:
504 0932 1 global data
505 0933 1
506 0934 1 Implicit Outputs:
507 0935 1 global data
508 0936 1
509 0937 1 Returned Value:
510 0938 1 none
511 0939 1
512 0940 1 Side Effects:
513 0941 1
514 0942 1 --
515 0943 1
516 0944 1
517 0945 2 global routine anl$object_hdr_text(record_number,the_record): novalue = begin
518 0946 2
519 0947 2 bind
520 0948 2 record_dsc = .the_record: descriptor;
521 0949 2
522 0950 2 own
523 0951 2 record_msg: vector[7,long] initial(
524 0952 2 0,
525 0953 2 anl$obj$_objlrmrec,
526 0954 2 anl$obj$_objsrcrec,
527 0955 2 anl$obj$_objttlrec,
528 0956 2 anl$obj$_objcprrec,
529 0957 2 0,
530 0958 2 anl$obj$_objgtxrec);
531 0959 2 local
532 0960 2 scanp: ref block[,byte],
533 0961 2 work_dsc: descriptor;
534 0962 2
535 0963 2
536 0964 2 ! First we print the main record line for this text record.
537 0965 2
538 0966 2 scanp = .record_dsc[ptr];
539 0967 2 anl$object_record_line(.record_msg[.scanp[obj$_b_subtyp]],record_number,record_dsc);
540 0968 2 anl$report_line(0);
541 0969 2
542 0970 2 ! Now we format the textual information into lines, with as many characters
543 0971 2 ! per line as possible. SCANP will act as the text pointer.
544 0972 2
545 0973 2 anl$format_line(0,1,anl$obj$_texthdr);
546 0974 2 scanp = .scanp + 2;
547 0975 2 while .scanp lssa (.record_dsc[ptr]+.record_dsc[len]) do (
548 0976 2
549 0977 2 ! Build a descriptor for this line of text.
```

```
0978 3
0979 build_descriptor(work_dsc,minu(.record_dsc[ptr]+.record_dsc[len]-.scanp,65)..scanp);
0980
0981 ! Print the text.
0982
0983 anl$format_line(0,1,anlobj$_text,.work_dsc[len],.work_dsc[ptr]);
0984
0985 ! Update the text pointer.
0986
0987 scanp = .scanp + .work_dsc[len];
0988 );
0989
0990 return;
0991
0992 1 end;
```

.PSECT \$OWNS\$,NOEXE,2

```
00000000 0000C RECORD_MSG:
00000000G 00000000G 00000000G 00000000G 00010 .LONG 0
00000000 00020 .LONG ANLOBJ$_OBJLNMREC, ANLOBJ$_OBJSRCREC, -
00000000G 00024 .LONG ANLOBJ$_OBJTTLREC, ANLOBJ$_OBJCPREC
00000000G 00024 .LONG 0
00000000G 00024 .LONG ANLOBJ$_OBJGTXREC
```

.PSECT \$CODE\$,NOWRT,2

```
000C 00000
5E 08 08 C2 00002 .ENTRY ANL$OBJECT_HDR_TEXT, Save R2,R3
52 08 AC D0 00005 SUBL2 #8, SP
53 04 A2 D0 00009 MOVL THE RECORD, R2
04 52 DD 0000D MOVL 4(R2), SCANP
01 AC DD 0000F PUSHL R2
50 01 A3 9A 00012 PUSHL RECORD NUMBER
0000G CF 0000'CF 40 DD 00016 MOVZBL 1(SCANP), R0
0000G CF 03 FB 00018 PUSHL RECORD MSG[R0]
0000G CF 7E D4 00020 CALLS #3, ANL$OBJECT_RECORD_LINE
0000G CF 01 FB 00022 CLRL -(SP)
00000000G 8F DD 00027 CALLS #1, ANL$REPORT_LINE
01 DD 0002D PUSHL #ANLOBJ$_TEXTHDR
7E D4 0002F PUSHL #1
0000G CF 03 FB 00031 CLRL -(SP)
53 02 C0 00036 CALLS #3, ANL$FORMAT_LINE
50 62 3C 00039 ADDL2 #2, SCANP
50 04 A2 C0 0003C ADDL2 (R2), R0
50 53 D1 00040 ADDL2 4(R2), R0
50 53 1E 00043 CMPL SCANP, R0
00000041 50 53 C2 00045 BGEQU 3$
8F 50 D1 00048 SUBL2 SCANP, R0
50 04 1B 0004F CMPL R0, #65
04 6E 41 50 D0 00051 BLEQU 2$
AE 53 D0 00058 MOVZBL #65, R0
MOVL R0, WORK_DSC
MOVL SCANP, WORK_DSC+4
```

OBJMISC - Analyze Miscellaneous Object Records	15-Sep-1984 23:42:42	VAX-11 Bliss-32 V4.0-742
ANLSOBJECT_HDR_TEXT - Analyze Text Header Recor	14-Sep-1984 11:52:57	[ANALYZ.SRC]OBJMISC.B32:1

Page 27
(8)

		04	AE	DD	0005C	PUSHL	WORK_DSC+4
7E		04	AE	3C	0005F	MOVZWL	WORK_DSC, -(SP)
	00000000G		8F	DD	00063	PUSHL	#ANLOBJ\$-TEXT
		01	DD	00069	PUSHL	#1	
		7E	D4	0006B	CLRL	-(SP)	
0000G	CF	05	FB	0006D	CALLS	#5, ANLSFORMAT_LINE	
	50	6E	3C	00072	MOVZWL	WORK_DSC, RO	
	53	50	C0	00075	ADDL2	RO, SCANP	
		BF	11	00078	BRB	1\$	
		04	0007A	38:	RET		

0983
0987
0975
0992

; Routine Size: 123 bytes, Routine Base: SCODES + 050C

```
566 0993 1 %sbttl 'ANL$OBJECT_HDR_MTC - Analyze Maintenance Header Records'
567 0994 1 **
568 0995 1 Functional Description:
569 0996 1 This routine is called to analyze maintenance header records.
570 0997 1
571 0998 1 Formal Parameters:
572 0999 1 record_number The number of this record in the object file.
573 1000 1 the_record The address of the descriptor of the record.
574 1001 1
575 1002 1 Implicit Inputs:
576 1003 1 global data
577 1004 1
578 1005 1 Implicit Outputs:
579 1006 1 global data
580 1007 1
581 1008 1 Returned Value:
582 1009 1 none
583 1010 1
584 1011 1 Side Effects:
585 1012 1
586 1013 1 --
587 1014 1
588 1015 1
589 1016 2 global routine anl$object_hdr_mtc(record_number,the_record): novalue = begin
590 1017 2
591 1018 2 bind
592 1019 2 record_dsc = .the_record: descriptor;
593 1020 2
594 1021 2 local
595 1022 2 status: long,
596 1023 2 scanp: ref block[,byte],
597 1024 2 fit_ok: byte,
598 1025 2 work_dsc: descriptor;
599 1026 2
600 1027 2
601 1028 2 ! We begin by printing a record line for this maintenance record.
602 1029 2
603 1030 2 anl$object_record_line(anlobj$_objmtcrec,.record_number,record_dsc);
604 1031 2 anl$report_line(0);
605 1032 2
606 1033 2 ! Now we print the patch utility name.
607 1034 2
608 1035 2 scanp = .record_dsc[ptr];
609 1036 2 fit_ok = true;
610 1037 2 ensure_ascii_fit(0,0,8,0,record_dsc,work_dsc);
611 1038 2 if .fit_ok then (
612 1039 2 anl$format_line(0,1,anlobj$_objmtcname,.work_dsc[len],.work_dsc[ptr]);
613 1040 2 scanp = .work_dsc[ptr] + .work_dsc[len];
614 1041 2 );
615 1042 2
616 1043 2 ! Next we print the patch utility version.
617 1044 2
618 1045 2 ensure_ascii_fit(0,0,8,0,record_dsc,work_dsc);
619 1046 2 if .fit_ok then (
620 1047 2 anl$format_line(0,1,anlobj$_objmtcversion,.work_dsc[len],.work_dsc[ptr]);
621 1048 2 scanp = .work_dsc[ptr] - .work_dsc[len];
622 1049 2 );
```



```
623 1050
624 1051 ! Now the UIC of the stupid patch person (WHY NOT JUST RECOMPILE?).
625 1052
626 1053 ensure field_fit(0,0,16,0,record_dsc);
627 1054 if .fit_ok then (
628 1055     anl$format_line(0,1,anlobj$_objmtcuic,.scanp[0,0,8,0],.scanp[1,0,8,0]);
629 1056     scanp = .scanp + 2;
630 1057 );
631 1058
632 1059 ! Now the input file specification.
633 1060
634 1061 ensure ascii_fit(0,0,8,0,record_dsc,work_dsc);
635 1062 if .fit_ok then (
636 1063     anl$format_line(0,1,anlobj$_objmtcinput,.work_dsc[ptr],.work_dsc[ptr]);
637 1064     scanp = .work_dsc[ptr] + .work_dsc[ptr];
638 1065 );
639 1066
640 1067 ! Now the correction file specification.
641 1068
642 1069 ensure ascii_fit(0,0,8,0,record_dsc,work_dsc);
643 1070 if .fit_ok then (
644 1071     anl$format_line(0,1,anlobj$_objmtccorrect,.work_dsc[ptr],.work_dsc[ptr]);
645 1072     scanp = .work_dsc[ptr] + .work_dsc[ptr];
646 1073 );
647 1074
648 1075 ! Now the date and time of patching.
649 1076
650 1077 ensure field_fit(0,0,17*8,0,record_dsc);
651 1078 if .fit_ok then (
652 1079     build_descriptor(work_dsc,17,.scanp);
653 1080     anl$format_line(0,1,anlobj$_objmtcwhen,work_dsc);
654 1081     anl$check_when(work_dsc);
655 1082     scanp = .scanp + 17;
656 1083 );
657 1084
658 1085 ! Last, and hopefully least, the sequence number.
659 1086
660 1087 ensure field_fit(0,0,8,0,record_dsc);
661 1088 if .fit_ok then (
662 1089     anl$format_line(0,1,anlobj$_objmtcseqnum,.scanp[0,0,8,0]);
663 1090     increment (.scanp);
664 1091 );
665 1092
666 1093 ! Finally, we ensure that there are no spurious bytes at the end.
667 1094
668 1095 if .record_dsc[ptr]+.record_dsc[ptr] gtru .scanp then
669 1096     anl$format_error(anlobj$_extrabytes);
670 1097
671 1098 return;
672 1099
673 1100 end;
```

58	0000G	CF	9E	00002	MOVAB	R8	
57	0000G	CF	9E	00007	MOVAB	ANLSFORMAT_LINE, R8	
56	00000000G	8F	D0	0000C	MOVAB	ANLSFORMAT_ERROR, R7	
5E		08	C2	00013	MOVL	#ANLOBS_FIELDFIT, R6	
54	08	AC	D0	00016	SUBL2	#8, SP	
		54	DD	0001A	MOVL	THE_RECORD, R4	1019
	04	AC	DD	0001C	PUSHL	R4	1030
	00000000G	8F	DD	0001F	PUSHL	RECORD_NUMBER	
0000G	CF	03	FB	00025	PUSHL	#ANLOBS_OBJMTCREC	
		7E	D4	0002A	CALLS	#3, ANLSOBJECT_RECORD_LINE	
0000G	CF	01	FB	0002C	CLRL	-(SP)	1031
55	04	A4	D0	00031	CALLS	#1, ANLSREPORT_LINE	
52		55	D0	00035	MOVL	4(R4), R5	1035
53		01	90	00038	MOVL	R5, SCANP	
78		53	E9	0003B	MOVB	#1, FIT_OK	1036
51	01	A2	9E	0003E	BLBC	FIT_OK, 3\$	1037
50		64	3C	00042	MOVAB	1(R2), R1	
50		55	C0	00045	MOVZWL	(R4), R0	
50		51	D1	00048	ADDL2	R5, R0	
		07	1B	0004B	CMPL	R1, R0	
		56	DD	0004D	BLEQU	1\$	
67		01	FB	0004F	PUSHL	R6	
		53	94	00052	CALLS	#1, ANLSFORMAT_ERROR	
6A		53	E9	00054	CLRB	FIT_OK	
6E		62	9A	00057	BLBC	FIT_OK, 4\$	
AE	01	A2	9E	0005A	MOVZBL	(SCANP), WORK_DSC	
7F		53	E9	0005F	MOVAB	1(R2), WORK_DSC+4	
50		6E	3C	00062	BLBC	FIT_OK, 5\$	
50		08	C6	00065	MOVZWL	WORK_DSC, R0	
51	01	A042	9E	00068	DIVL2	#8, R0	
50		64	3C	0006D	MOVAB	1(R0)[SCANP], R1	
50		55	C0	00070	MOVZWL	(R4), R0	
50		51	D1	00073	ADDL2	R5, R0	
		07	1B	00076	CMPL	R1, R0	
		56	DD	00078	BLEQU	2\$	
67		01	FB	0007A	PUSHL	R6	
		53	94	0007D	CALLS	#1, ANLSFORMAT_ERROR	
7E		53	E9	0007F	CLRB	FIT_OK	
	04	AE	DD	00082	BLBC	FIT_OK, 6\$	1038
7E	04	AE	3C	00085	PUSHL	WORK_DSC+4	1039
	00000000G	8F	DD	00089	MOVZWL	WORK_DSC, -(SP)	
		01	DD	0008F	PUSHL	#ANLOBS_OBJMTCNAME	
		7E	D4	00091	PUSHL	#1	
68		05	FB	00093	CLRL	-(SP)	
52		6E	3C	00096	CALLS	#5, ANLSFORMAT_LINE	
52	04	AE	C0	00099	MOVZWL	WORK_DSC, SCANP	1040
79		53	E9	0009D	ADDL2	WORK_DSC+4, SCANP	
51	01	A2	9E	000A0	BLBC	FIT_OK, 7\$	1045
50		64	3C	000A4	MOVAB	1(R2), R1	
50		55	C0	000A7	MOVZWL	(R4), R0	
50		51	D1	000AA	ADDL2	R5, R0	
		07	1B	000AD	CMPL	R1, R0	
		56	DD	000AF	BLEQU	3\$	
67		01	FB	000B1	PUSHL	R6	
		53	94	000B4	CALLS	#1, ANLSFORMAT_ERROR	
7A		53	E9	000B6	CLRB	FIT_OK	
					BLBC	FIT_OK, 8\$	

04	6E		62	9A	000B9	MOVZBL	(SCANP), WORK_DSC	
	AE	01	A2	9E	000BC	MOVAB	1(R2), WORK_DSC+4	
	6F		53	E9	000C1	BLBC	FIT_OK, 8\$	
	50		6E	3C	000C4	MOVZWL	WORK_DSC, R0	
	50		08	C6	000C7	DIVL2	#8, R0	
	51	01	A042	9E	000CA	MOVAB	1(R0)[SCANP], R1	
	50		64	3C	000CF	MOVZWL	(R4), R0	
	50		55	C0	000D2	ADDL2	R5, R0	
	50		51	D1	000D5	CMPL	R1, R0	
			07	1B	000D8	BLEQU	5\$	
			56	DD	000DA	PUSHL	R6	
67			01	FB	000DC	CALLS	#1, ANLSFORMAT_ERROR	
			53	94	000DF	CLRB	FIT_OK	
73			53	E9	000E1	BLBC	FIT_OK, 10\$	1046
	04		AE	DD	000E4	PUSHL	WORK_DSC+4	1047
7E	04		AE	3C	000E7	MOVZWL	WORK_DSC, -(SP)	
	00000000G		8F	DD	000EB	PUSHL	#ANL0BJ\$_OBJMTCVERSION	
			01	DD	000F1	PUSHL	#1	
			7E	D4	000F3	CLRL	-(SP)	
			05	FB	000F5	CALLS	#5, ANLSFORMAT_LINE	
68			6E	3C	000F8	MOVZWL	WORK_DSC, SCANP	1048
52	04		52	C3	000FB	SUBL3	SCANP, WORK_DSC+4, SCANP	
			53	E9	00100	BLBC	FIT_OK, 11\$	1053
		02	A2	9E	00103	MOVAB	2(R2), R1	
			64	3C	00107	MOVZWL	(R4), R0	
			55	C0	0010A	ADDL2	R5, R0	
			51	D1	0010D	CMPL	R1, R0	
			07	1B	00110	BLEQU	7\$	
			56	DD	00112	PUSHL	R6	
67			01	FB	00114	CALLS	#1, ANLSFORMAT_ERROR	
			53	94	00117	CLRB	FIT_OK	
79			53	E9	00119	BLBC	FIT_OK, 12\$	1054
	01		A2	9A	0011C	MOVZBL	1(SCANP), -(SP)	1055
7E			62	9A	00120	MOVZBL	(SCANP), -(SP)	
	00000000G		8F	DD	00123	PUSHL	#ANL0BJ\$_OBJMTCUIC	
			01	DD	00129	PUSHL	#1	
			7E	D4	0012B	CLRL	-(SP)	
			05	FB	0012D	CALLS	#5, ANLSFORMAT_LINE	
68			02	C0	00130	ADDL2	#2, SCANP	1056
52			53	E9	00133	BLBC	FIT_OK, 13\$	1061
78			53	E9	00133	BLBC	FIT_OK, 13\$	
51	01		A2	9E	00136	MOVAB	1(R2), R1	
			64	3C	0013A	MOVZWL	(R4), R0	
			55	C0	0013D	ADDL2	R5, R0	
			51	D1	00140	CMPL	R1, R0	
			07	1B	00143	BLEQU	9\$	
			56	DD	00145	PUSHL	R6	
67			01	FB	00147	CALLS	#1, ANLSFORMAT_ERROR	
			53	94	0014A	CLRB	FIT_OK	
			53	E9	0014C	BLBC	FIT_OK, 14\$	
6A			62	9A	0014F	MOVZBL	(SCANP), WORK_DSC	
04		01	A2	9E	00152	MOVAB	1(R2), WORK_DSC+4	
			53	E9	00157	BLBC	FIT_OK, 15\$	
			6E	3C	0015A	MOVZWL	WORK_DSC, R0	
			08	C6	0015D	DIVL2	#8, R0	
		01	A042	9E	00160	MOVAB	1(R0)[SCANP], R1	
			64	3C	00165	MOVZWL	(R4), R0	
			55	C0	00168	ADDL2	R5, R0	

OBJMISC
V04-000OBJMISC - Analyze Miscellaneous Object Records 15-Sep-1984 23:42:42
ANLSOBJECT_HDR_MTC - Analyze Maintenance Header 14-Sep-1984 11:52:57VAX-11 Bliss-32 V4.0-742
[ANALYZ.SRC]OBJMISC.B32;1Page 32
(9)

50		51	D1	0016B	CMPL	R1, R0	
		07	1B	0016E	BLEQU	11\$	
		56	DD	00170	PUSHL	R6	
67		01	FB	00172	CALLS	#1, ANLSFORMAT_ERROR	
		53	94	00175	CLRB	FIT_OK	
7D		53	E9	00177	BLBC	FIT_OK, 16\$	1062
	04	AE	DD	0017A	PUSHL	WORK_DSC+4	1063
7E	04	AE	3C	0017D	MOVZWL	WORK_DSC, -(SP)	
	00000000G	8F	DD	00181	PUSHL	#ANL0BJ\$_OBJMTCINPUT	
		01	DD	00187	PUSHL	#1	
		7E	D4	00189	CLRL	-(SP)	
68		05	FB	0018B	CALLS	#5, ANLSFORMAT_LINE	
52		6E	3C	0018E	MOVZWL	WORK_DSC, SCANP	1064
52	04	AE	C0	00191	ADDL2	WORK_DSC+4, SCANP	
78		53	E9	00195	BLBC	FIT_OK, 17\$	1069
51	01	A2	9E	00198	MOVAB	1(R2), R1	
50		64	3C	0019C	MOVZWL	(R4), R0	
50		55	C0	0019F	ADDL2	R5, R0	
50		51	D1	001A2	CMPL	R1, R0	
		07	1B	001A5	BLEQU	13\$	
		56	DD	001A7	PUSHL	R6	
67		01	FB	001A9	CALLS	#1, ANLSFORMAT_ERROR	
		53	94	001AC	CLRB	FIT_OK	
5F		53	E9	001AE	BLBC	FIT_OK, 17\$	
6E		62	9A	001B1	MOVZBL	(SCANP), WORK_DSC	
AE	01	A2	9E	001B4	MOVAB	1(R2), WORK_DSC+4	
77		53	E9	001B9	BLBC	FIT_OK, 18\$	
50		6E	3C	001BC	MOVZWL	WORK_DSC, R0	
50		0B	C6	001BF	DIVL2	#8, R0	
51	01	A042	9E	001C2	MOVAB	1(R0)[SCANP], R1	
50		64	3C	001C7	MOVZWL	(R4), R0	
50		55	C0	001CA	ADDL2	R5, R0	
50		51	D1	001CD	CMPL	R1, R0	
		07	1B	001D0	BLEQU	15\$	
		56	DD	001D2	PUSHL	R6	
67		01	FB	001D4	CALLS	#1, ANLSFORMAT_ERROR	
		53	94	001D7	CLRB	FIT_OK	
70		53	E9	001D9	BLBC	FIT_OK, 19\$	1070
	04	AE	DD	001DC	PUSHL	WORK_DSC+4	1071
7E	04	AE	3C	001DF	MOVZWL	WORK_DSC, -(SP)	
	00000000G	8F	DD	001E3	PUSHL	#ANL0BJ\$_OBJMTCORRECT	
		01	DD	001E9	PUSHL	#1	
		7E	D4	001EB	CLRL	-(SP)	
68		05	FB	001ED	CALLS	#5, ANLSFORMAT_LINE	
52		6E	3C	001F0	MOVZWL	WORK_DSC, SCANP	1072
52	04	AE	C0	001F3	ADDL2	WORK_DSC+4, SCANP	
67		53	E9	001F7	BLBC	FIT_OK, 20\$	1077
51	11	A2	9E	001FA	MOVAB	17(R2), R1	
50		64	3C	001FE	MOVZWL	(R4), R0	
50		55	C0	00201	ADDL2	R5, R0	
50		51	D1	00204	CMPL	R1, R0	
		07	1B	00207	BLEQU	17\$	
		56	DD	00209	PUSHL	R6	
67		01	FB	0020B	CALLS	#1, ANLSFORMAT_ERROR	
		53	94	0020E	CLRB	FIT_OK	
4E		53	E9	00210	BLBC	FIT_OK, 20\$	1078
6E		11	D0	00213	MOVL	#17, WORK_DSC	1079

OBJMISC
V04-000

OBJMISC - Analyze Miscellaneous Object Records 15-Sep-1984 23:42:42
ANLSOBJECT_HDR_MTC - Analyze Maintenance Header 14-Sep-1984 11:52:57

VAX-11 Bliss-32 V4.0-742
[ANALYZ.SRC]OBJMISC.B32;1

Page 33
(9)

04	AE	52	DD 00216	MOVL	SCANP, WORK_DSC+4	
		5E	DD 0021A	PUSHL	SP	1080
	00000000G	8F	DD 0021C	PUSHL	#ANLOBJ\$_OBJMTCWHEN	
		01	DD 00222	PUSHL	#1	
		7E	D4 00224	CLRL	-(SP)	
	68	04	FB 00226	CALLS	#4, ANLSFORMAT_LINE	
		5E	DD 00229	PUSHL	SP	1081
0000G	CF	01	FB 0022B	CALLS	#1, ANLSCHECK_WHEN	
	52	11	CO 00230	ADDL2	#17, SCANP	1082
	2B	53	E9 00233	BLBC	FIT_OK, 20\$	1087
	51	A2	9E 00236	MOVAB	1(R2), R1	
	50	64	3C 0023A	MOVZWL	(R4), R0	
	50	55	CO 0023D	ADDL2	R5, R0	
	50	51	D1 00240	CMPL	R1, R0	
		07	1B 00243	BLEQU	19\$	
		56	DD 00245	PUSHL	R6	
	67	01	FB 00247	CALLS	#1, ANLSFORMAT_ERROR	
		53	94 0024A	CLRB	FIT_OK	
	12	53	E9 0024C	BLBC	FIT_OK, 20\$	1088
	7E	62	9A 0024F	MOVZBL	(SCANP), -(SP)	1089
		8F	DD 00252	PUSHL	#ANLOBJ\$_OBJMTCSEQNUM	
		01	DD 00258	PUSHL	#1	
		7E	D4 0025A	CLRL	-(SP)	
	68	04	FB 0025C	CALLS	#4, ANLSFORMAT_LINE	
		52	D6 0025F	INCL	SCANP	1090
	50	64	3C 00261	MOVZWL	(R4), R0	1095
	50	55	CO 00264	ADDL2	R5, R0	
	52	50	D1 00267	CMPL	R0, SCANP	
		09	1B 0026A	BLEQU	21\$	
	00000000G	8F	DD 0026C	PUSHL	#ANLOBJ\$_EXTRABYTES	1096
	67	01	FB 00272	CALLS	#1, ANLSFORMAT_ERROR	
		04	00275	21\$:	RET	1100

; Routine Size: 630 bytes, Routine Base: \$CODE\$ + 0587

; 674 1101 1

21
6)OBJMISC
V04-000OBJMISC - Analyze Miscellaneous Object Records
ANL\$OBJECT_LNK - Analyze LNK RecordK 3
15-Sep-1984 23:42:42
14-Sep-1984 11:52:57VAX-11 Bliss-32 V4.0-742
[ANALYZ.SRC]OBJMISC.B32;1Page 34
(10)

```

: 676 1102 1 %sbttl 'ANL$OBJECT_LNK - Analyze LNK Record'
: 677 1103 1 ++
: 678 1104 1 Functional Description:
: 679 1105 1 This routine analyzes the LNK record, with link option specifications.
: 680 1106 1 Currently this is ignored by the linker, so we will just dump it in
: 681 1107 1 hex for the guy.
: 682 1108 1
: 683 1109 1 Formal Parameters:
: 684 1110 1 record_number The number of this object record.
: 685 1111 1 the_record Address of descriptor of record.
: 686 1112 1
: 687 1113 1 Implicit Inputs:
: 688 1114 1 global data
: 689 1115 1
: 690 1116 1 Implicit Outputs:
: 691 1117 1 global data
: 692 1118 1
: 693 1119 1 Returned Value:
: 694 1120 1 none
: 695 1121 1
: 696 1122 1 Side Effects:
: 697 1123 1
: 698 1124 1 --
: 699 1125 1
: 700 1126 1
: 701 1127 2 global routine anl$object_lnk(record_number,the_record): novalue = begin
: 702 1128 2
: 703 1129 2 bind
: 704 1130 2 record_dsc = .the_record: descriptor;
: 705 1131 2
: 706 1132 2
: 707 1133 2 ! First we print a major line for the record.
: 708 1134 2
: 709 1135 2 anl$object_record_line(anlobj$_objlnkrec,.record_number,record_dsc);
: 710 1136 2 anl$report_line(0);
: 711 1137 2
: 712 1138 2 ! Now we just dump the contents in hex.
: 713 1139 2
: 714 1140 2 anl$format_hex(1,record_dsc);
: 715 1141 2
: 716 1142 2 return;
: 717 1143 2
: 718 1144 1 end;
```

```

                                0000 0000
                                7E      04 AC 7D 00002
                                0000G CF 00000000G 8F DD 00006
                                0000G CF              03 FB 0000C
                                0000G CF              7E D4 00011
                                0000G CF              01 FB 00013
                                0000G CF              08 AC DD 00018
                                0000G CF              01 DD 0001B
                                0000G CF              02 FB 0001D
```

```

.ENTRY ANL$OBJECT_LNK, Save nothing
MOVQ RECORD_NUMBER, -(SP)
PUSHL #ANLOBJ$_OBJLNKREC
CALLS #3, ANL$OBJECT_RECORD_LINE
CLRL -(SP)
CALLS #1, ANL$REPORT_LINE
PUSHL THE_RECORD
PUSHL #1
CALLS #2, ANL$FORMAT_HEX
```

```

: 1127
: 1135
:
: 1136
: 1140
:
```

OBJMISC
V04-000

OBJMISC - Analyze Miscellaneous Object Records
ANL\$OBJECT_LNK - Analyze LNK Record

15-Sep-1984 23:42:42
14-Sep-1984 11:52:57

VAX-11 Bliss-32 V4.0-742
[ANALYZ.SRC]OBJMISC.B32;1

Page 35
(10)

04 00022

RET

: 1144

; Routine Size: 35 bytes, Routine Base: \$CODE\$ + 07FD

; 719 1145 1
; 720 1146 0 end eludom

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	40 NOVEC, WRT, RD	,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$PLITS	12 NOVEC,NOWRT, RD	,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$CODE\$	2080 NOVEC,NOWRT, RD	, EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	37	0	1000	00:01.9

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:OBJMISC/OBJ=OBJ\$:OBJMISC MSRC\$:OBJMISC/UPDATE=(ENH\$:OBJMISC)

; Size: 2080 code + 52 data bytes
; Run Time: 00:36.0
; Elapsed Time: 01:49.3
; Lines/CPU Min: 1912
; Lexemes/CPU-Min: 17522
; Memory Used: 290 pages
; Compilation Complete

0007 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

OB MISC
LIS

RM521DX
LIS

RM531DX
LIS

RM5
LIS

OB TTR
LIS